1998

Computer-Aided Mechanical Design Using Configuration Spaces

Leo Joskowicz

Elisha Sacks
Purdue University, eps@cs.purdue.edu

Report Number:
98-027

COMPUTER-AIDED MECHANICAL DESIGN USING CONFIGURATION SPACES

Leo Joskowicz
Elisha Sacks

CSD-TR #98-027
August 1998

Computer-Aided Mechanical Design Using Configuration Spaces

Leo Joskowicz
Institute of Computer Science
The Hebrew University
Jerusalem 91904, Israel
E-mail: josko@cs.huji.ac.il

Elisha Sacks (corresponding author)
Computer Science Department
Purdue University
West Lafayette, IN 47907, USA
E-mail: eps@cs.purdue.edu

August 27, 1998

Abstract

The paper describes research in computer-aided design of mechanical systems using configuration spaces. The research addresses the core design task of rigid-body contact analysis and related tasks. Contact analysis is a computational bottleneck in mechanical design, especially in systems with complex part shapes, tight fits, and changing contacts. Manual analysis is error-prone and time-consuming, whereas automated analysis exceeds the capabilities of current design software. To address these problems, we have developed a general contact analysis method for planar mechanical systems based on configuration space computation. We have implemented a prototype design environment that integrates contact analysis with simulation, tolerance analysis, and visualization. The software helps designers study system function under a range of operating conditions, find and correct design flaws, and optimize performance.

Keywords: computer-aided design, mechanism theory, contact analysis, kinematics.
Submitted to IEEE Computational Science and Engineering.

Version: 2

1 Introduction

This paper describes our research in computer-aided design of mechanical systems using configuration spaces. Mechanical design is the task of devising an assembly of parts (a mechanical system) that performs a function reliably and economically. It is a ubiquitous activity with applications in mechanical, electrical, and biomedical engineering. Designers need to devise, analyze, and compare competing design prototypes to create a good design. Computer-aided design helps designers reduce design time and improve design quality by replacing physical prototypes with electronic ones.

Our research addresses the core design task of rigid-body contact analysis and related design tasks. We assume that the parts are rigid: they cannot change shape or overlap. This assumption is reasonable for most mechanical design tasks. Contact analysis determines the positions and orientations at which the parts of a system touch and the ways that the touching parts interact. The interactions consist of constraints on the part motions that prevent them from overlapping. The constraints are expressed as algebraic equations that relate the part coordinates. For example, a round ball on a flat floor obeys the constraint $z - r = 0$ with z the height of its center point (a position coordinate) and r its radius. The constraints are a function of the shapes of the touching part features (vertices, edges, and faces), hence they change when one pair of features breaks contact and another makes contact.

Contact analysis is a core design task because contacts are the physical primitives that make mechanical systems out of collections of parts. Systems perform functions by transforming motions via part contacts. The shapes of the interacting parts impose constraints on their motions that largely determine the system function. Designers perform contact analysis to derive this evolving sequence of contacts and motion constraints. The results help them simulate the system function, find and correct design flaws, measure performance, and compare design alternatives.

We illustrate contact analysis and its role in design on the ratchet mechanism shown in Figure 1. The mechanism has four moving parts and a fixed frame. The driver, link, and ratchet are attached to the frame by revolute joints. (A revolute joint is a cylindrical pin on one part that fits in a matching cylindrical hole in the other part, thus restricting the part motions to rotation about the cylinder axis.) The pawl is attached to the link by a revolute joint and is attached to a torsional spring (not shown) that applies a counterclockwise torque around the joint. A motor rotates the driver with constant angular velocity, causing the link pin to move left and right. This causes the link to oscillate around its rotation point, which moves the pawl left and right. The leftward motion pushes a ratchet tooth, which rotates the ratchet counterclockwise. The rightward motion frees the pawl tip from the tooth, which allows the spring to rotate the pawl to engage the next tooth.

Contact analysis validates the intended function by determining if the link oscillates far enough, if the pawl pushes the ratchet teeth far enough, if the system can jam, and so on. Revolute joints are standard and easy to analyze, although the combined effect of several joints is complex. The driver/link pair is harder to analyze because the link pin interacts with the inner and outer driver profiles. The ratchet/pawl pair is much harder yet because the part shapes are complex and because

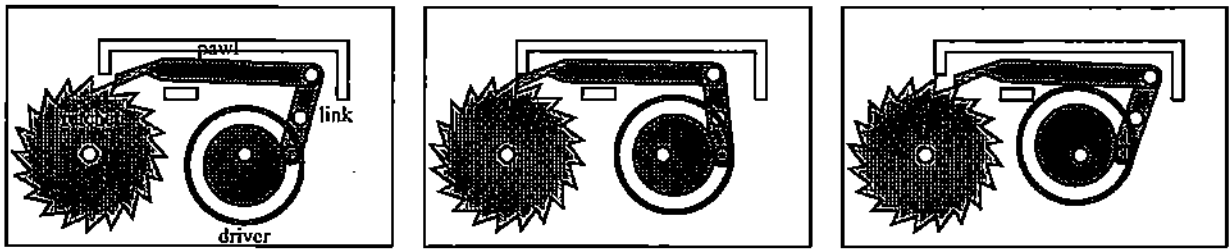


Figure 1: Ratchet mechanism: (a) pawl advancing ratchet; (b) pawl fully advanced; (c) pawl retracting. White circles indicate revolute joints.

the pawl can translate horizontally, translate vertically, and rotate, whereas the other parts only rotate. Contact analysis of the overall system is hardest of all because it must validate the intended interactions among all the parts, such as the indirect relation between the driver and the ratchet, and must rule out interference, such as the pawl hitting the frame.

Contact analysis is a computational bottleneck in mechanical systems with many potential part contacts. The complexity grows rapidly as the number of parts increases. A pair of touching parts interacts via contacts between feature pairs. Hundreds of features per part is the norm, which leads to thousands of potential contacts per pair and to a combinatorial growth in system contacts. Our research shows that this complexity is common in modern mechanisms. Contact changes occur by design in 65% of the 2500 mechanisms in an engineering encyclopedia [1]. Examples include gears, cams, ratchets and clutches. Manufacturing variation often introduces unintended, complex contacts into a system whose nominal function has simple contacts, such as revolute joints with play, where the pin the pin can translate as well as rotate because it is smaller than the hole. Designers need to analyze these variations to ensure correct function. This process is called tolerance analysis.

Manual contact analysis of complex systems is error-prone and time-consuming, whereas automated analysis exceeds the capabilities of current design software. This software consists mainly of simulators for systems whose parts interact via standard joints, such as linkage mechanisms and robot manipulators [2]. It cannot handle the 65% of systems with non-standard joints and contact changes.

To address these problems, we have developed a general contact analysis method for planar mechanical systems. Planar systems account for over 90% of mechanisms. We have implemented a prototype design environment that integrates contact analysis with simulation, tolerance analysis, and visualization. The software helps designers study system function under a range of operating conditions, find and correct design flaws, and optimize performance.

The contact analysis method is based on configuration space computation. Configuration space is a geometric representation of rigid body interaction that has seen extensive computational use in robot motion planning [3]. We have found that it is an effective tool for contact analysis. The

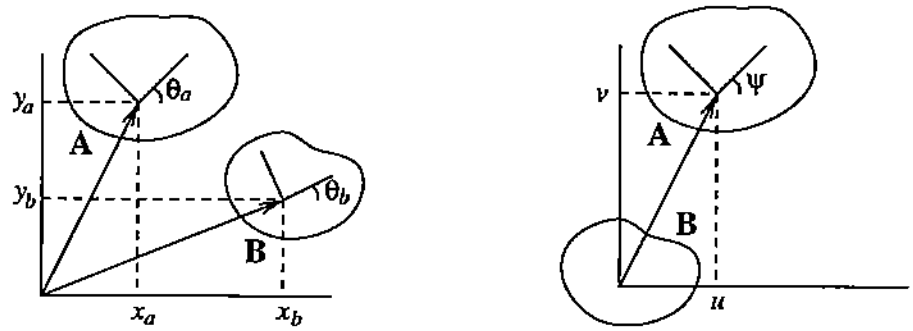


Figure 2: Pairwise configurations: (a) absolute coordinates and (b) relative coordinates.

configuration space of a mechanical system describes all possible part interactions. It encodes quantitative information, such as part motion paths, and qualitative information, such as system failure modes. It provides a framework within which diverse design tasks can be performed. We have developed a configuration space computation algorithm for systems with curved parts, generalized it to toleranced parts, and applied it to mechanical design.

This paper is a survey of our research on contact analysis. We describe the configuration space representation, explain its value as a contact analysis tool, and illustrate it on simple examples. In so doing, we present the big picture behind the diverse algorithms that appear in prior publications. We assess the strengths and weaknesses of our approach for practical design tasks and identify future research issues.

2 Configuration space

We perform contact analysis on a mechanical system by computing a configuration space for each pair of parts. The purpose of the analysis is to identify the pairs of part features that touch in some system configuration, to compute the motion constraints for every pair, and to compute the configurations in which contacts change. This section explains what configuration space is and how it supports contact analysis.

We attach reference frames to the parts and define the configuration of a part to be the position and orientation of its reference frame with respect to a fixed global frame. Figure 2a) shows two parts A and B, their reference frames, and their configurations (x_a, y_a, θ_a) and (x_b, y_b, θ_b). The configuration space of the pair is the Cartesian product, $(x_a, y_a, \theta_a, x_b, y_b, \theta_b)$, of the part configurations. The configuration space coordinates represent the six independent motions of the parts, called degrees of freedom. As the parts move, the configuration traces a path in configuration space.

Configuration space partitions into three disjoint sets that characterize part interaction: blocked

space where the parts overlap, free space where they do not touch, and contact space where they touch without overlap. The free and blocked spaces are open sets whose common boundary is contact space. This implies that the first two have the same dimension as the configuration space, whereas the dimension of the third is one lower. Intuitively, free and blocked space are open because disjoint or overlapping parts remain so under all small motions, whereas contact space is closed because touching parts separate or overlap under some small motions.

We illustrate these concepts with a simple example: a block that moves in a fixed frame (Figure 3a). The frame is fixed at the global origin with orientation 0, so we can drop its coordinates from the configuration space and consider only the block coordinates (u, v, ψ). We first assume that the block translates in the displayed orientation without rotating, which yields a two-dimensional configuration space. The gray region is blocked space, the white region is free space, and the black lines are contact space. The dot in free space marks the displayed position of the block. Free space divides into a central rectangle where the block is inside the frame, an outer region where it is outside, and a narrow connecting rectangle where it is partly inside. The contact curves (lines in this case) bounding these regions represent contacts between the vertices and edges of the block and the frame. Changing the orientation of the block yields configuration spaces with different topologies (Figure 3b, c). The free space consists of disconnected inside and outside regions because the block does not fit through the frame mouth.

We now consider the same example, but with the block orientation a variable. The configuration space becomes three dimensional with rotation coordinate ψ (Figure 4). One way to visualize this space is as a stack of planar slices along the rotation axis. Each slice is the configuration space of a block that translates at a fixed orientation, such as the three examples above. The full space is the union of the slices. The free space consists of an outer tube, an inner tube, and two connecting channels near $\psi = \pm\pi/2$ where the block is nearly vertical. The outer tube is the union of the outer regions, the inner tube is the union of the inner rectangles, and the channels are the union of the connecting regions. Blocked space is the region between the tubes and outside the channels.

We can model general planar pairs with three-dimensional configuration spaces even though they have six degrees of freedom. The reason is that the part contacts are invariant under rigid motions of the pair. In other words, the relative configuration of the parts determines the contacts. We compute the configuration space of part a with respect to the reference frame of part b, which is equivalent to fixing b in the $(0, 0, 0)$ configuration. The relation between the absolute and relative coordinate systems is given by

$$\begin{aligned} u &= \cos\theta_b(x_a - x_b) + \sin\theta_b(y_a - y_b) \\ v &= \cos\theta_b(y_a - y_b) - \sin\theta_b(x_a - x_b) \\ \psi &= \theta_a - \theta_b. \end{aligned} \tag{1}$$

Figure 2 contrasts the two coordinate systems. In the example, the block is a, the frame is b, and $(u, v, \psi) = (x_a, y_a, \theta_a)$ because $x_b, y_b, \theta_b = 0$.

Whatever its dimension, the configuration space of a pair is a complete representation of the part contacts, so any contact question is answerable by a configuration space query. Testing if

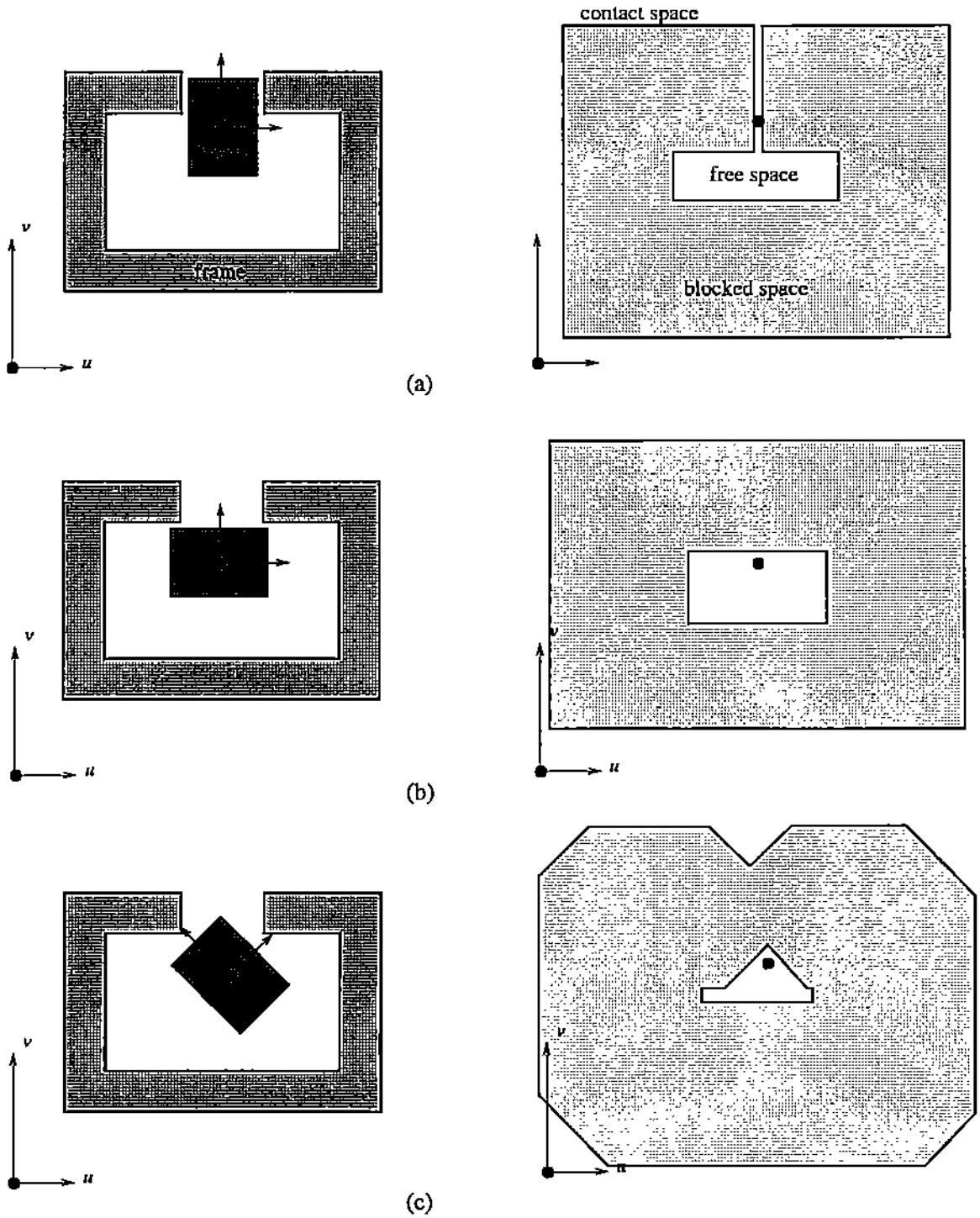


Figure 3: A translating block moving around a fixed frame.

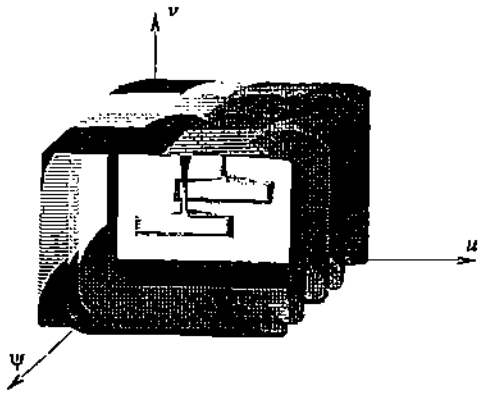


Figure 4: Configuration space for block that translates and rotates. The orientation ψ ranges from $-\pi$ to π. Free and blocked space are white. Contact space is shaded with a unique color for each contact patch.

parts overlap, do not touch, or are in contact in a given configuration corresponds to testing if the configuration point is in blocked, free, or contact space. Contacts between pairs of features correspond to contact patches (curve segments in two dimensions and surface patches in three). The patch geometry encodes the motion constraint and the patch boundary encodes the contact change conditions. Part motions correspond to paths in configuration space. A path is legal if it lies in free and contact space, but illegal if it intersects blocked space. Contacts occur at configurations where the path crosses from free to contact space, break where it crosses from contact to free space, and change where it crosses between neighboring contact patches.

The configuration space representation generalizes from pairs of parts to systems with more than two parts. A system of n planar parts has a $3n$-dimensional configuration space whose points specify the n part configurations. A system configuration is free when no parts touch, is blocked when two parts overlap, and is in contact when two parts touch and no parts overlap. System configuration spaces allow us to analyze multi-part interactions, such as the motion relation between the driver and the ratchet in the ratchet mechanism. They are difficult to compute in general, but we have developed a practical algorithm for systems of planar, one degree of freedom parts [1]. We discuss them no further in this paper.

3 Configuration space computation

Configuration space computation has algebraic and combinatorial components. The algebraic task is to derive the contact constraint for a pair of features and to compute the resulting contact patch. The computational complexity is proportional to the product of the degrees of the features. The combinatorial task is to compute the configuration space partition, which is determined by contact patch intersections. This is a computational geometry problem, called arrangement computation,

and is solvable in nearly linear time (in the number of feature contacts) for planar configuration spaces and in quadratic time for three-dimensional configuration spaces.

The robotics literature contains many configuration space computation algorithms, most of which appear in Latombe [3]. That research provides practical algorithms for pairs of polygons. These algorithms do not extend to curved parts because they rely on the special structure of polygonal contact spaces, which are made up of ruled surface patches generated by vertex/edge contacts. Polygons are fine for path planning, which is the primary robotics application of configuration space, but are inappropriate for mechanical design where precise motion constraints between curved features are crucial to system function.

We have developed a fast, robust configuration space computation algorithm for pairs of planar parts whose boundaries are comprised of line segments and of circular arcs. These features suffice for most engineering applications. The program distinguishes between standard joints, fixed-axes pairs, and general pairs. The standard joint types are revolute, prismatic, and sliding. We have seen revolute joints in the ratchet mechanism; the other types are described in engineering texts. Each standard joint imposes a permanent contact that induces a fixed set of motion constraints, which the program retrieves from a table. Fixed-axes pairs consists of two parts with one degree of freedom apiece. They are analyzed by a very fast algorithm that exploits their special structure. These two types of pairs account for over 90% of planar pairs based on our survey of 2500 mechanisms [1]. The remaining pairs are analyzed by a general algorithm.

Fixed-axes pairs have two-dimensional configuration spaces whose coordinates are the motion coordinates of the two parts. For example, Figure 5 shows the configuration space of the driver/link pair. The configuration space coordinates are the driver orientation θ and the link orientation ω. The upper and lower contact curves represent contacts between the cylindrical pin and the outer and inner cam profiles. The free space is the region in between. As the driver rotates from $\theta = -\pi$ to $\theta = 0$, its inner profile pushes the link pin right, which rotates the link counter-clockwise from $\omega = -0.47$ radians to $\omega = 0.105$ radians. As the driver rotates from $\theta = 0$ to $\theta = \pi$, the pin breaks contact with the inner profile and makes contact with the outer one, which pulls it left and rotates the link clockwise. The configuration follows the lower contact curve from $\theta = -\pi$ to 0, travels horizontally through free space until it hits the upper contact curve, and follows it to π.

The fixed-axes program [4] computes these two-dimensional configuration spaces. The contact curves are obtained from a hand-computed table with one entry for each combination of feature types (line segments, arcs, and points) and motion types (horizontal translation, vertical translation, and rotation), for example a rotating line segment and a circle translating horizontally. The program enumerates the feature pairs, generates their contact curves from the table, and computes the configuration partition with a planar line sweep algorithm. It handles any realistic pair in well under one second (100,000 contacts in 0.1 seconds on a workstation).

The general program [5] computes three-dimensional configuration spaces. The contact patches are implicit, the patch boundary curves are parametric, and the contact space is in boundary representation. The patches and curves are obtained from a hand-computed table as before. The partition is computed with a planar line sweep by dimension reduction. The running time is under

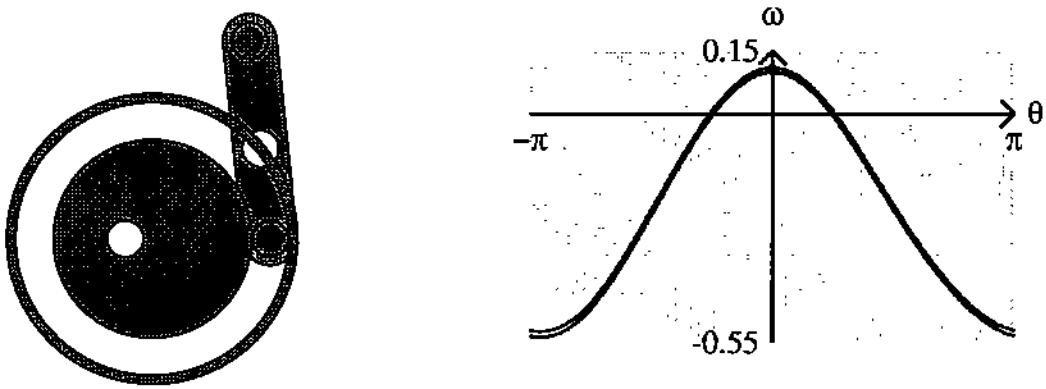
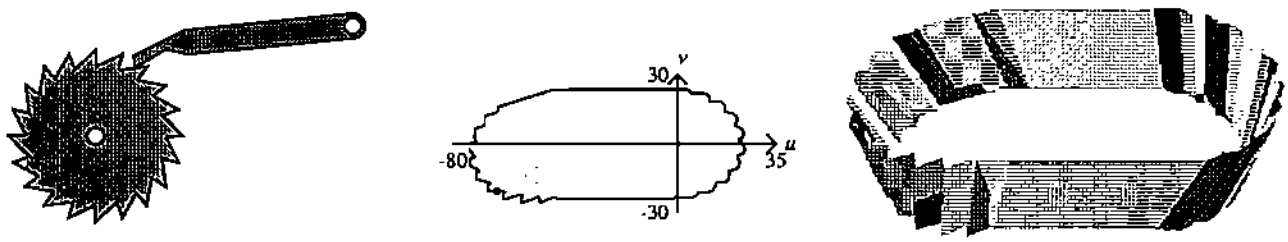


Figure 5: Driver/link pair and its configuration space.



(a) (b) (c)

Figure 6: (a) Ratchet/pawl pair; (b) configuration space; (b) slice at $\psi = 0.277$ radians.

one minute for every pair that we have tested, including ones with 10,000 contacts.

Figure 6 shows the three-dimensional configuration space of the ratchet/pawl pair. Although hard to visualize, it encodes a full contact analysis. Let us examine the slice in the figure, which shows how the parts translate in the displayed orientations. The dot marks the displayed position of the ratchet relative to the pawl. It lies on a contact curve that represents contact between the pawl tip and the side of a ratchet tooth. The right end of the curve is the intersection point with a second contact curve that represents contact between the left corner of the pawl and the next tooth counter-clockwise. The ratchet can maintain this contact while translating right until the second contact occurs and further translation is blocked. Rotation can only be expressed in the full configuration space.

4 Dynamical simulation

Configuration spaces support a novel form of dynamical simulation that is well-suited to mechanical design. Dynamical simulation means computing the motions of the parts in accordance with Newton's laws. It is an important design tool for those aspects of system function that depend on

dynamical effects, such as inertia, friction, and gravity. It provides loads for finite-element analysis: a numerical analysis of part interactions that lie outside the rigid-body idealization, such as deformation and stress.

We illustrate simulation on the ratchet mechanism. The external forces are a motor that rotates the driver, a torsional spring that rotates the pawl counterclockwise, and the weight of the pawl. A short simulation shows that the first cycle appears correct (Figure 1). But a longer simulation reveals a problem: the ratchet rotates faster at every cycle because of the repeated pawl impacts. We correct this problem by adding an external load or internal damping to the ratchet. Further simulation tests how strong a spring is needed to maintain the pawl/ratchet contact with various driving torques.

Contact analysis is a prerequisite for simulation because contacts create forces that effect part motion. The simulator needs to know which part features touch at every instant and when contact changes occur. Given this information, it can compute the contact forces, combine them with the external forces, compute the part accelerations from Newton's laws, and numerically integrate them to obtain the part configurations and velocities at the next time step. At each time step, the simulator checks for contact changes since the previous step, which force it to back up to the change time and update the contact equations. Manual analysis is practical for systems with permanent contacts, called multi-body systems, and is implemented in commercial simulators [6], but automated analysis is crucial for systems with many contact changes.

We have developed a simulator that uses configuration spaces for automated contact analysis [7]. The configuration spaces of the interacting pairs are computed before the simulation. At each time step, the simulator queries them for the contact data for contact force computation. It tests for part collisions and contact changes between steps by querying the configuration spaces for transitions between free and contact space or between contact patches. We have simulated systems with tens of moving parts at interactive speeds. The largest simulation to date is a chain assembly with two gears, a 34-link chain connected by pin joints, and 68 link/gear higher pairs.

An alternate approach, developed primarily in graphics research [8], is to test all pairs of parts for collisions at each time step. The test can be extremely fast even for large systems. The disadvantages for mechanical systems are that current algorithms do not handle curved parts and are inefficient when the parts are close together and interact often. The main advantage is that the algorithm handles three-dimensional parts.

5 Tolerance analysis

Configuration spaces support automated tolerance analysis of mechanical systems. The task is to compute the variation in the system function due to manufacturing variation in the parts. If we use parametric part models, the part variations can be represented as intervals around the nominal parameter values. In the ratchet mechanism, the parameters would include the driver radius and eccentricity, the link length, the pawl length and tip angle, and the ratchet tooth length and slant.

The analysis can be qualitative, quantitative, or statistical. Qualitative analysis tells us if the part parameter variations can cause unintended contact effects, for example if the pawl can hit the frame or can fail to advance the ratchet. This analysis must be performed first because it provides the contact constraints for the other types. Quantitative analysis gives us the derivatives of the part configurations with respect to the parameters, which allows us to compute the maximal error in system function. Statistical analysis estimates the fraction of mechanisms that will fail.

Tolerance analysis presupposes contact analysis because the variation in the system function arises from variations in the part contact constraints. We need to know which contacts occur at each stage of the work cycle and how their constraints depend on the part parameters. Manual analysis is often infeasible because of the many contacts and the complex relations between part parameters and contact constraints. We [9] have developed a tolerance analysis algorithm for planar systems based on a generalization of configuration space to parametric parts. The algorithm performs quantitative and statistical analyses and helps designers perform qualitative analysis. It analyzes systems with 50 to 100 parameters in a few minutes, which permits interactive tolerancing of detailed functional models.

Figure 7 shows a generalized configuration space for the driver/link pair of the ratchet mechanism: colored curves superimposed on the nominal configuration space. The red and green curves are upper and lower bounds on the variation in the contact space due to the part variations. If the part parameters are in their tolerance intervals, the contact space must be between these curves. The channel between the top green curve and the bottom red curve represents the worst-case pin clearance. It is smallest at $\theta = 0$ where the pin is at its rightmost position and largest where the pin is at its leftmost position. Increasing the part tolerances brings these curves closer. When they meet, the the qualitative mechanism function alters (a failure mode) because the pin cannot complete its cycle.

6 Conclusion

We have seen that configuration space computation is a practical algorithm for contact analysis of planar mechanical systems. Our design software is accurate and fast based on thousands of complex test cases.

The next step in our contact analysis research is configuration space computation for pairs of three-dimensional parts. The relative configuration space is six-dimensional because a part has three translations and three rotations. Computing it is much more difficult than the planar case: there are more features, the contact equations have higher degree, the rotations have a non-Euclidean geometry, and the computational geometry involves six dimensions. Prior research does not provide a practical algorithm for polyhedra, much less for curved parts. We believe that a general solution is impossible, so we will focus on specialized algorithms for important classes of pairs. We are working on the first, and most important class: pairs of parts that rotate about fixed axes.

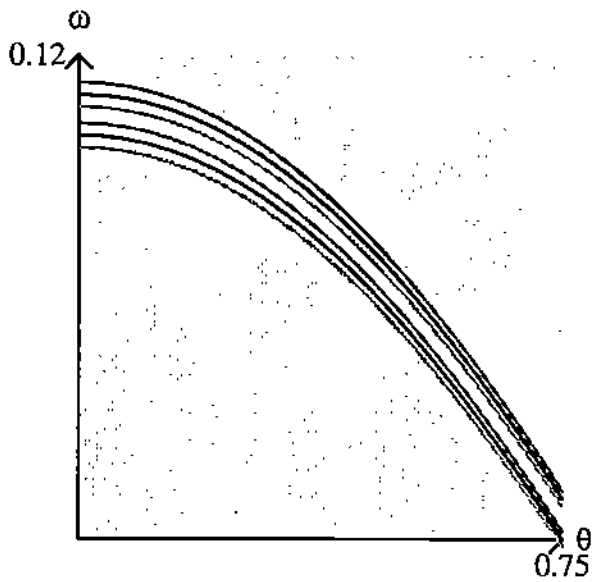


Figure 7: Detail of driver/link generalized configuration space.

The next step in our mechanical design research is to apply configuration spaces to industrial problems. We are working on automative transmission design with Ford Motors and on micro-mechanism design with Sandia National Laboratory. Both teams feel that our software will help them design faster and better. The next year should show if they are right. Ease of use will be crucial. We are developing an interactive, window based interface and visualization tools to meet this need.

Acknowledgments

Joskowicz is supported in part by a grant from the Authority for Research and Development, The Hebrew University and by a Guastalla Faculty Fellowship, Israel. Sacks is supported in part by NSF grants CCR-9617600 and CCR-9505745, by a gift from Ford Motors, and by the Purdue Center for Computational Image Analysis and Scientific Visualization.

References

[1] Leo Joskowicz and Elisha Sacks. Computational kinematics. *Artificial Intelligence*, 51:381–416, 1991. reprinted in [10].

[2] W. Schiehlen. *Multibody systems handbook*. Springer-Verlag, 1990.

[3] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[4] Elisha Sacks and Leo Joskowicz. Computational kinematic analysis of higher pairs with multiple contacts. *Journal of Mechanical Design*, 117:269–277, 1995.

[5] Elisha Sacks. Practical sliced configuration spaces for curved planar pairs. *International Journal of Robotics Research*, 1998. to appear.

[6] Edward J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, volume I: Basic Methods. Simon and Schuster, 1989.

[7] Elisha Sacks and Leo Joskowicz. Dynamical simulation of planar systems with changing contacts using configuration spaces. *Journal of Mechanical Design*, 120:181–187, 1998.

[8] Ming C. Lin, Dinesh Manocha, Jon Cohen, and Stefan Gottschalk. Collision detection: Algorithms and applications. In Jean-Paul Laumond and Mark Overmars, editors, *Algorithms for Robotic Motion and Manipulation*. A. K. Peters, Boston, MA, 1997.

[9] Elisha Sacks and Leo Joskowicz. Parametric kinematic tolerance analysis of general planar systems. *Computer-Aided Design*, 30(9):707–714, 1998.

[10] K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors. *The Algorithmic Foundations of Robotics*. A. K. Peters, Boston, MA, 1995.